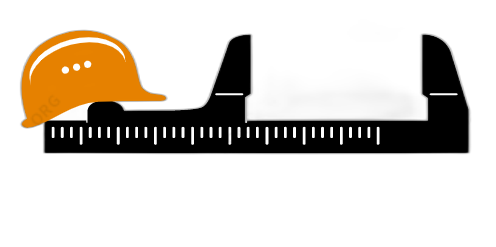
**ACME-TOOLKITS**

**GROUP WIS ARCHITECTURE TESTING KNOWLEDGE**



ACME - TOOLKITS

Repositorio: <https://github.com/framurpri/Acme-Toolkits.git>

Fecha: 02/02/2022

Miembros del grupo:

* Fernando Andres Galindo
* Francisco Murillo Prior
* Daniel Rodríguez Camacho
* Alvaro Rodríguez García
* Antonio Rosado Barrera
* Jose Manuel Ruiz Perez

**ÍNDICE**

[RESUMEN EJECUTIVO](#_rtba4hi5hexy) 2

1. [Introducción](#_34yagubucmqr) 3
2. [Conocimientos de testing](#_dn7pg6fz5ddv) 3
3. [Conclusiones](#_w1x1fw317ixk) 3
4. [Bibliografía](#_ec7t27h464zy) 4

## **RESUMEN EJECUTIVO**

En este documento redactamos los conocimientos que tiene el grupo sobre el testing de la arquitectura WIS obtenidos en las asignaturas previas que hemos cursado. Para ello hemos realizado una reunión donde cada uno ha expuesto sus ideas y aquí es donde se recogen y unifican.

Control de versión del documento:

| Versión del documento | Fecha | Descripción |
| --- | --- | --- |
| 1.0 | 25/02/2022 | Creación del documento |
| 1.1 | 26/02/2022 | Añadido todo el contenido y formato |
| 1.2 | 01/03/2022 | Añadido link del repositorio |

# Introducción

El proceso de testing en el mundo del desarrollo y la informática es muy importante ya que te ahorra mucho trabajo aunque sea una parte trabajosa. Debemos mantener una integración contínua y a ser posible de forma automática. Para ello el grupo debe conocer bien cómo se procede y qué conocimientos son necesarios para realizar los diferentes tipos de test. En conjunto hemos unificado de forma general los conocimientos del grupo sobre este mundo para saber donde hay que hacer más hincapié a la hora del aprendizaje y conocer los puntos fuertes y débiles en este aspecto.

# Conocimientos de testing

Los conocimientos que tenemos en este ámbito los hemos obtenido de diversas asignaturas que se listan a continuación con cada punto que hemos practicado en ellas.

* **Arquitectura e Integración de Sistemas Software**: en esta asignatura tuvimos una primera toma de contacto con el testing pero sólo de forma teórica, aprendiendo los diferentes tipos de testing como por ejemplo las pruebas unitarias o las de integración.
* **Diseños y pruebas I:** en esta asignatura profundizamos en los test unitarios mediante JUnit 5. Utilizando el framework de Java Spring Boot tuvimos que hacer test sobre las clases, los métodos y los controladores.

Para ello las pruebas más comunes que usamos fueron del tipo AssertEquals y AssertThrows.

* **Procesos de Software y Gestión:** aprendimos la importancia de los test automatizados y de la integración contínua cuando se trabaja en equipo pero nuevamente de forma teórica.

# Conclusiones

Como conclusión obtenemos que la mayoría del grupo sabe sobre los test unitarios pero no sobre el resto por lo que habrá que estudiar cuáles vamos a necesitar y profundizar en ellos. Además el grupo conoce bien la herramienta de JUnit 5 para hacer testing en Spring Boot.

Sobre la integración contínua los conocimientos están un poco más crudos por lo que deberíamos poner más interés en ese apartado.

Añadir que el rol de tester solo lo tienen dos miembros del grupo pero a la hora de alguna duda o problema el resto del grupo tiene que estar al tanto y ser capaz de poder ayudar en cualquier circunstancia necesaria.

# Bibliografía

Intencionadamente en blanco